QPREP SUBMIT COMMAND

To submit a job to PBS, the following command is used:

$ qprep myscript

where *myscript* is a file that contains the *qprep* options for the given job.

DIRECTIVE SYNTAX

Options to qprep are given in the form of key=value pairs called directives. In the preamble of a job script, directives have the form

#PSTQ key=value

and the preamble ends with the line

#PSTQ END_OF_PREAMBLE.

Typically, the first line in the script is the specific shell under which the script should be executed. For example,

#!/bin/ksh

would run the script under ksh.

QPREP CONTROL DIRECTIVES

Control the behavior of the qprep while translating a script.

Directive	Options	Description
silent	y	Do not write jobid to standard out.
out_script	Output file	Write the translated script to this file.
submit	y|n	Submit the job after translating.
verbose_script	y|n	Include extra comments about the translation.
operation_verbosity	n	Set the verbosity level of comments.

EXTRA DIRECTIVES

Pass the string to the target queue submission line just before the name of the script with

#PSTQ target_cmd_line_args=string

Pass the string to the output script, preceded by the target directive syntax (which is usually just a dash) with

#PSTQ target_script_directive=string

JOB ENVIRONMENT DIRECTIVES

Control the environment in which the qprep script operates.

Directive	Options	Description
account	Account name	Charge the specified account name.
username	User name	Run the job as the specified user.
export_vars	y|n	Export the environment.
shell	Shell name	The shell used to run the script.

REPORTING DIRECTIVES

Control standard output, error, and mail reporting.

Directive	Options	Description
stderr	File	Define standard output file.
stdout	File	Define standard error file.
keep	stderr,stdout	Keep either or both.
mail	a,b,e,r,t	Send mail when a: aborted b: begins e: end r: rerun t: routed
mail_to	User1[,User2,…]	List of addresses.

JOB CONTROL DIRECTIVES

Specify how and where to run a job.

Directive	Options	Description
queue	Name	Name of the queue.
parallel_env	Name	Name of the parallel environment.
job_name	Name	Name of the job.
checkpoint	n|s|number	Checkpointing characteristics n: never s: when the queue is shutdown number: every so many minutes

LIMIT DIRECTIVES

Specify resource limits to run a job. Note that all times are to be specified with hh:mm.

Directive	Options	Description
nodes	Number	Number of nodes to be used.
cpus_per_node	Number	Number of CPUs per node to be used.
total_cpus	Number	Total number of cpus to be used.
wall_time	Time	Maximum wall-clock time.
process_cpu_time	Time	Maximum time used by a single process.
total_cpu_time	Time	Total time used by all processes.
proc_file_size	Size (Mbytes)	Maximum size of files written by a single process.
total_file_size	Size (Mbytes)	Maximum size of all files written.
process_memory	Size (Mbytes)	Maximum memory used by a single process.
total_memory	Size (Mbytes)	Maximum memory used by all processes.

JOB DEPENDENCY DIRECTIVES

Specify the dependencies on other jobs.

Directive	Options	Description
sync_count	Number	Sync number of jobs where number is the first job.
sync_with	Jobid	Sync jobid when *sync_count* is set.
after	Jobid[,Jobid,…]	Execute after these jobs have begun.
after_ok	Jobid[,Jobid,…]	Execute after these jobs terminate successfully.
after_not_ok	Jobid[,Jobid,…]	Execute after these jobs terminate with an error.
after_any	Jobid[,Jobid,…]	Execute after these jobs regardless of success.
depends_on	Number	Execute after the dependencies below.
before	Jobid[,Jobid,…]	Jobs can be run after the execution begins.
before_ok	Jobid[,Jobid,…]	Jobs can be run after the current job terminates successfully.
before_not_ok	Jobid[,Jobid,…]	Jobs can be run after the current job terminates with an error.
before_any	Jobid[,Jobid,…]	Jobs can be run after the current job terminates regardless of success.